

PERANCANGAN DAN ANALISIS SISTEM PEMANTUAN PADA TETESAN INFUS MELALUI NOTIFIKASI PADA LINE BOT.

DESIGN AND ANALYSIS OF MONITORING SYSTEM ON DRIP CHAMBER THROUGH NOTIFICATION ON THE LINE BOT.

Adinda Ayu Herdianto¹Dr. Ir. Rendy Munadi, M.T²Gustommy Bisono, S.T., M.T³

Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

Adindaherdianto@gmail.comRendymunadi@telkomuniversity.ac.idBisono@telkomuniversity.ac.id

ABSTRAK

Penelitian ini mengajukan suatu gagasan untuk mendeteksi tetesan infus dengan sistem informasi melalui Line yang akan dikirim ke perawat yang berada di ruangan perawat maupun di ruangan lainnya, perawat dapat mengetahui lebih cepat kondisi cairan infus setiap pasien di ruangan perawatan. Dengan hasil pengujian performansi yang penulis uji, menghasilkan minimnya terjadi *Packet Loss* pada komunikasi antara Arduino dengan CloudMQTT Broker yaitu senilai 0,01%. Untuk menghitung performansi jaringan, penulis mendapatkan hasil rata-rata *Delay* sebesar 35,512 ms pada saat pengiriman 3 pengguna, Sementara untuk hasil pada pengiriman lebih dari 10 pengguna mendapatkan nilai 104,65 ms. Hasil rata-rata *Throughput* sebesar 15513,593 bps pada pengiriman dengan 3 pengguna. Sementara untuk hasil pengiriman lebih dari 10 pengguna mendapatkan nilai 4978,092 bps.

Selain itu penulis, mengukur keandalan dan kesiapan alat dengan pengujian pengambilan data 30 kali setiap 15 menit dan secara acak, Dengan hasil pengujian keandalan pada sistem ini senilai 87% pada keandalan sistem mendeteksi pergerakan tetesan infus dan 83% pada pengiriman status infus pada *user*, sedangkan kesiapan alat memiliki hasil 100%.

Kata Kunci : *Infus, NodeMCU, CloudMQTT, Heroku, Line*

ABSTRACT

This study proposes an idea to detect infusion droplets with information systems through the Line that will be sent to nurses who are in the nurse's room as well as in other rooms, nurses can find out more quickly the infusion fluid conditions for each patient in the treatment room. With the results of the performance testing that the author tested, resulting in minimal occurrence of Packet Loss in communication between Arduino and CloudMQTT Brokers, which is worth 0.01 To calculate network performance, the authors get an average delay of 35,512 ms when sending 3 users, while for results on sending more than 10 users get a value of 104.65 ms. The average throughput of 15513,593 bps on delivery with 3 users. As for the results of sending more than 10 users get a value of 4978,092 bps.

In addition, the authors measure the reliability and availability of tools by testing data retrieval 30 times every 15 minutes and randomly, with the results of reliability testing on this system worth 87% on the reliability of the system to detect infusion droplets and 83% on sending infusion status to the user, while the availability of tools has a 100% yield.

Keywords : *Infusion, NodeMCU, CloudMQTT, Heroku, Line*

1. Pendahuluan

Alat kedokteran sebagai fungsi diagnostik dan terapi sangat berkaitan dengan aspek keselamatan dan keamanan. Infus cairan intravena merupakan sebuah jarum yang masuk ke dalam pembuluh vena untuk memberikan sejumlah cairan ke dalam tubuh yang digunakan untuk menggantikan cairan tubuh yang hilang. Cairan yang dimasukkan dengan menggunakan alat ini langsung melalui pembuluh darah, sehingga cairan ini dapat diolah tubuh secara langsung. Infus ini biasanya diberikan ketika seseorang mengalami pendarahan dalam jumlah yang banyak, trauma abdomen yang berat dan juga ketika sudah mengalami sakit diare dan demam cukup lama.

Setiap rumah sakit atau puskesmas (Pusat Kesehatan Masyarakat) pasti menggunakan infus dari mulai penyakit ringan hingga yang berat. Bahkan saat ini, banyak orang yang memilih untuk menggunakan infus daripada menggunakan obat-

obat alami, bahkan ketika hanya kelelahan sedikit lebih memilih untuk di infus dibandingkan untuk melakukan cara alami tubuh seperti minum sekitar 8 gelas per harinya agar memenuhi cairan tubuh. Semakin tingginya penggunaan infus di rumah sakit, membuat perawat menjadi sedikit kewalahan dalam menangani beberapa pasien. Pada saat ini, sistem pemantauan hanya berpusat di meja perawat.

Maka dari itu, pada penelitian kali ini penulis menggunakan perkembangan teknologi yang ada untuk membuat alat yang dapat memberikan sistem pemantauan terhadap masalah yang dapat terjadi pada saat menggunakan infus ini sendiri. Pada salah satu jurnal [1] membuat penulis mengembangkan teknologi ini. Pada jurnal tersebut memiliki kekurangan dalam pendeteksian tetesan air sehingga data kurang akurat. Pada referensi tersebut tidak memilikinya sistem pengingat dan hanya dapat melihat kondisi infus tersebut.

Penulis melakukan studi literatur terhadap permasalahan terkait pada jurnal tersebut, dan menemukan jurnal pendukung untuk membantu proses pengiriman data nantinya, terutama pada penggunaan Smart Things, seperti pada jurnal [2] menggunakan protokol MQTT. Pada jurnal [3] menggunakan protokol yang sama pada pendeteksian infus tanpa memberi tahu kepada perawat membuat penulis menambahkan beberapa fitur yaitu pemberian notifikasi dengan aplikasi Line sebagai media untuk pemantau infus pasien kepada perawat.

Menggunakan arduino yang memiliki sensor inframerah untuk mendeteksi tetesan infus hingga pendeteksian jumlah cairan yang dibutuhkan dan dibatasi. Serta menggunakan sistem monitoring yang memudahkan perawat yang menjaga untuk mengetahui kondisi tetesan infus pada pasien dengan aplikasi Line. Line sudah banyak digunakan untuk berbagai sosial media dengan banyak layanan yang sudah dapat di akses pada aplikasi ini. Dengan harapan nantinya alat dan sistem pengingat ini bermanfaat dan dapat menyempurnakan penelitian sebelumnya.

2. Konsep Dasar

2.1 Intravena Infus

Infus cairan intravena merupakan sebuah jarum yang masuk ke dalam pembuluh vena untuk memberikan sejumlah cairan ke dalam tubuh yang digunakan untuk menggantikan cairan tubuh yang hilang.

Pada umumnya infus diberikan kepada pasien yang butuh penanganan obat dengan cepat dan juga ketika pasien tidak dapat menerima atau tidak diperkenankan untuk memasukan apapun lewat mulutnya. Bisa juga ketika pasien sedang mengalami muntah-muntah, sesak nafas, epilepsi atau kejang-kejang.

Tujuan utama dari pemberian infus ini adalah reaksi obat yang cepat di absorpsi dari pada injeksi yang lainnya, mencegah kerusakan jaringan dan yang terakhir untuk memasukan obat dalam jumlah yang lebih besar. Pemberian infus ini sendiri dapat dilakukan pada aliran darah vena *Medianan Cubitus* (Lengan), vena *Saphenous* (Tungkai), vena *Jugularis* (Leher), vena *Frontalis/Temporal* pada bagian kepala.

2.2 Internet of Things

Internet of Things menyediakan kemampuan bagi manusia dan komputer agar dapat berinteraksi dari miliaran hal dari sensor, aktuator, layanan dan objek lain yang terhubung ke internet. Terwujudnya IoT ini dapat mengubah dan memberdayakan interaksi manusia dengan dunia. Dalam realisasi sistem IoT adalah *Middleware*, *Middleware* biasa di gambarkan sebagai perangkat lunak yang dirancang untuk menjadi perantara dalam perangkat dan aplikasi IoT agar memudahkan penggunaan.

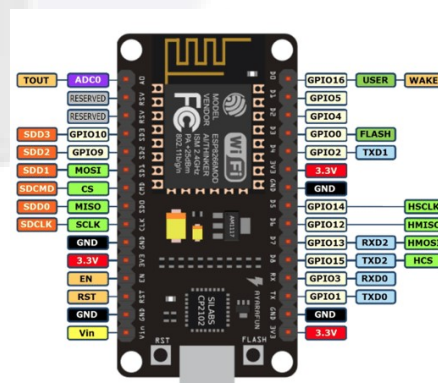
Pada dasarnya *Internet of Things* memungkinkan para penggunaannya untuk memvisualisasi, menjelajahi dan membangun analisis pada *cloud computing* yang ada. Aplikasi yang digunakan diharapkan dapat melakukan interaksi secara real time. Pada saat itu, Kevin Ashton membangun sistem sensor dengan identifikasi frekuensi radio (RFID) yang digunakan untuk menghubungkan dunia fisik dengan internet. Dengan asal mulanya itu, menjadikan dasar pengembangan IoT ini untuk meranah ke berbagai bidang seperti rumah, kendaraan, perawatan, industri, toko ritel dan banyak lagi.

Perangkat yang nantinya akan digunakan, berkomunikasi melalui berbagai macam protokol yang nantinya dirancang untuk menoleransi tidak stabilnya koneksi dan mengurangi kebutuhan lebar pita jaringan. Semua komunikasi IoT harus diamankan seperti menggunakan identitas perangkat, enkripsi dan kontrol akses serta pemantauan perangkat. Pada akhirnya diharapkan teknologi IoT ini nantinya memiliki sistem yang memperkuat secara mandiri dan memperbaiki secara mandiri [9].

2.3 Arduino

Arduino adalah pengendali mikro *single-board* yang bersifat *open source* diturunkan dari Wiring Platform [10]. Pada dasarnya tujuan awal dibuatnya arduino dibuat agar para pelajar dapat membuat perangkat dengan modal yang sedikit dan mudah digunakan. Keuntungan dari Arduino ini sendiri adalah harganya yang terjangkau, dapat dijalankan pada berbagai sistem operasi seperti windows, linux, Mac, dan sebagainya. Lalu arduino sendiri memiliki bahasa pemrograman yang mudah untuk dipelajari dan open source jadi, tidak diutamakan untuk anak teknik juga bahkan orang awam pun dapat menggunakan dengan mudah.

2.3.1 ESP8266



Gambar 2.1 Modul Sheet NodeMCU ESP8266

ESP8266 dijadikan modul WiFi yang digemari para *hardware developer*. Memiliki keuntungan utama yaitu harga yang sangat terjangkau dengan fungsi serbaguna membuat

ESP8266 menjadi favorit. ESP8266 memiliki sifat SoC (*System On Chip*), yang dibuat agar pengguna dapat melakukan pemrograman tanpa memerlukan mikrokontroler tambahan. ESP8266 juga dapat dijadikan sebagai Adhoc akses poin maupun klien.

Untuk pemrogramannya, dapat menambahkan *ESPLorer* untuk *firmware* berbasis NodeMCU dan menggunakan Putty sebagai *Terminal Control* untuk *AT Command*. Tidak lupa untuk menambahkan *library* ESP8266 pada *board manager* pada Arduino IDE nantinya.

2.3.2 Modul Sensor IR Obstacle Avoidance

Modul IR *Obstacle Avoidance* merupakan modul yang berfungsi sebagai pendeteksi objek yang ada, terdiri dari inframerah dan photodiode. Inframerah menggunakan prinsip pantulan cahaya inframerah sebagai penentu nilainya dan menyesuaikan dengan sensitivitas yang sudah diatur pada intensitas cahayanya. Hasil yang akan didapat nantinya dapat masuk ke MCU, agar dapat melakukan kontrol terhadap device lain.

2.3.3 Arduino IDE

Arduino IDE adalah perangkat lunak yang digunakan untuk pemrograman sintaks fungsi-fungsi yang akan digunakan pada arduino yang nantinya digunakan. Arduino menggunakan bahasa pemrograman yang memiliki kesamaan dengan bahasa C, yang disebut *Sketch*. Saat ini *Sketch*, diubah agar memudahkan pemula dalam melakukan pemrograman. IC Mikrokontroler Arduino sebelum diperjual belikan diwajibkan untuk ditanamkan program *Bootlader* yang memiliki fungsi penengah antara *compiler* Arduino dengan mikrokontroler. Arduino IDE dibuat menggunakan bahasa pemrograman JAVA yang dilengkapi dengan *Library C/C++*.

2.4 MQTT

MQTT adalah *Message Queuing Telemetry Transport*, dimana MQTT adalah sebuah protokol yang bekerja diatas stack TCP/IP dan mempunyai ukuran paket data dengan *low overhead* yang kecil (minimum 2 bytes) sehingga berefek pada konsumsi catu daya yang kecil [12]. MQTT merupakan protokol *Data-Agnostic* yang nantinya dapat mengirimkan data, seperti *data binary*, text, XML ataupun JSON. MQTT memakai model *Publish/Subscribe* daripada model *Client-Server*. Keuntungan menggunakan model tersebut karna adanya *Space Decoupling* yang artinya bahwa antar sumber dan penerima tidak mengetahui adanya broker dan juga dengan adanya *Time Coupling* sumber dan penerima tidak perlu melakukan hubungan secara bersamaan karna adanya broker ini.

2.5 Heroku

Heroku merupakan sebuah *cloud platform* yang dapat menjalankan beberapa bahasa pemrograman seperti Ruby, Node.js, Python, Java,

PHP dan lainnya. Heroku termasuk ke dalam kriteria *Platform As A Service* sehingga pengguna dapat mengembangkan, menjalankan dan mengelola aplikasi tanpa kompleksitas membangun dan memelihara infrastruktur yang biasanya terkait dengan pengembangan dan peluncuran aplikasi.

Heroku memiliki beberapa manfaat yang sangat menguntungkan bagi penggunanya seperti heroku dapat menjalankan *script app* tanpa memerlukan konfigurasi yang rumit dan membuat pengembang aplikasi lebih fokus pada kode dan aplikasi mereka tanpa perlu memusingkan arsitektur dan server.

2.6 Line Bot

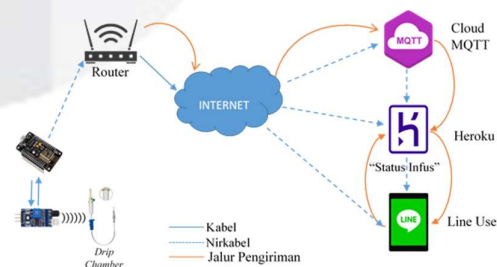
Line adalah sebuah platform komunikasi yang tidak hanya menawarkan layanan pesan, pesan suara dan video. Line juga merupakan *smart portal* dengan beragam solusi terkait konten, hiburan dan bisnis. Pengguna Line di Indonesia sudah memasuki angka 90% dari semua penduduk, dan meningkat pesat pada tahun 2014 hingga 2016. Pada bulan September 2016 Line mengeluarkan *Messaging API* dan diganti oleh BOT API karna formatnya lebih sederhana dan mudah dipelajari serta digunakan. Pada BOT API memiliki fitur seperti Konfirmasi, tombol, dan carousel.

Pada pesan konfirmasi disini pesan dapat memilih ya atau tidak. Pada pesan bertipe tombol dapat mengkombinasikan teks, gambar ataupun tautan. Pada pesan carousel dapat menampilkan lebih dari satu pesan dan dapat digeser ke kiri maupun ke kanan.

3. Perancangan Alat

3.1 Desain Sistem

Pada bab ini penulis akan menjelaskan tentang desain yang digunakan untuk memenuhi terwujudnya alat pendeteksian infus beserta parameter yang dijadikan bahan analisis nantinya. Dengan sistem seperti yang ditujukan pada gambar 3.1 yang nantinya akan di implementasikan.



Gambar 3. 1 Desain Sistem Alat pendeteksi Infus

Dalam sebuah rumah sakit, pasien yang menggunakan infus dapat terbilang cukup banyak dengan penggunaan infus yang dapat dimobilisasi oleh pasien itu sendiri. Dengan banyaknya infus yang digunakan maka perlu adanya sebuah server

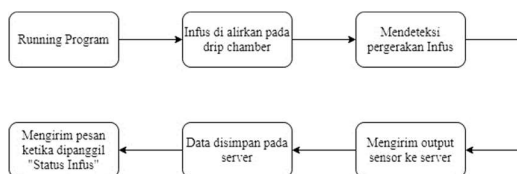
terpusat yang dapat digunakan untuk memantau penggunaan infus. ESP8266 yang sudah dilengkapi dengan modul dan sudah di konfigurasi sebagaimana

Kategori Delay	Besar Delay
Excellent	<150 ms
Good	150-300 ms
Poor	300-450 ms
Unnacceptable	>450 ms

mestinya akan di hubungkan dengan Heroku dimana nantinya menjadi tampilan utama pada pc yang ada, dan akan di deploy dengan BOT Line khusus perawat.

3.2 Sistem Perancangan Alat

Pada perancangan alat pendeteksi infus ini penulis akan merancang sistem monitoring pada tampilan pesan Line Bot menggunakan Line Developer. Pada bagian alatnya akan dipasang NodeMCU ESP8266 dan nantinya alat tersebut akan mengirimkan data pada jaringan yang disediakan, seperti pada gambar 3.2.



Gambar 3. 2 Blok Diagram Perancangan Alat

Pada awalnya NodeMCU dan sensor IR akan dipasang dan di hubungkan dengan USB pada komputer. Saat sudah terhubung, NodeMCU akan mencari koneksi internet yang telah disediakan dengan SSID yang sudah di konfigurasi. Setelah NodeMCU terhubung, pastikan server Postman pada Cloud MQTT dan Heroku Dyno Bot sudah dalam kondisi Running. Jika tidak terhubung maka NodeMCU harus dipastikan sudah terhubung dengan internet.

Sensor akan mulai mendeteksi dan mengirimkan hasil keluaran yang telah dibaca sensor menuju server Postman yang telah disediakan oleh Cloud MQTT. Pada saat sensor sudah terhubung, maka Websocket pada Cloud MQTT akan menunjukkan bahwa pesan topic yang dibawa oleh NodeMCU sudah diterima dan disimpan. Jika pesan tidak diterima, maka harus dipastikan bahwa NodeMCU memiliki akses kode yang sesuai dengan Postman pada Cloud MQTT.

Selanjutnya, Heroku Dyno Bot yang sudah Online akan menerima pengiriman topic, dan akan sampai pada pengguna ketika pengguna mengirim pesan "Status Infus". Jika pengguna hanya melihat huruf "R" pada tampilan line maka Heroku Dyno Bot kemungkinan tidak menyala. Skema Pengujian Sistem

3.3 Pengujian pada sistem pengolahan data

Pada tugas akhir ini, pengujian pada performansi jaringan QoS(Quality of Service) meliputi *Delay* dan *Throughput*, serta *Availability* dan *Reliability*. Penulis menggunakan aplikasi wireshark untuk menguji performansi jaringan yang ada. Ketentuan performansi jaringan *Delay* yang digunakan berpacu pada standarisasi ITU-T G.1010, seperti berikut :

Tabel 3. 1 Standarisasi ITU-T G.1010

Pengujian performansi *Throughput* berguna untuk mengetahui kecepatan pengiriman data dalam satuan *byte per second*. *Throughput* menggunakan rumus yaitu :

$$\text{Throughput} = \frac{\text{Juml data yang dikirim}}{\text{Waktu pengiriman data}} \quad (3.1)$$

Suatu sistem memiliki kemungkinan berhasil atau tidaknya dalam menjalankan kondisi yang diberikan oleh pembuat. Maka dari itu penulis akan menganalisis *Availability* dan *Reliability* performansi sistem. *Reliability* merupakan kemungkinan pada suatu sistem untuk memenuhi fungsi tertentu pada kondisi dan waktu yang ditentukan, dapat dirumuskan sebagai berikut :

$$\text{Reliability} = \frac{(\text{Uptime} - \text{Downtime})}{\text{Uptime}} \quad (3.2)$$

Tidak jauh beda dari *Reliability*, *Availability* merupakan kemungkinan pada suatu sistem pada saat sistem siap untuk beroperasi pada waktu yang dibutuhkan, dapat dirumuskan sebagai berikut :

$$\text{Availability} = \frac{\text{Uptime}}{\text{Uptime} + \text{Downtime}} \quad (3.3)$$

Uptime merupakan total waktu dalam kondisi suatu sistem dinyatakan berhasil, dan *Downtime* merupakan total waktu dalam kondisi gagal.

3.4 Analisis

Analisis yang dilakukan pada tugas akhir ini berfokus pada performansi jaringan berdasarkan QoS yaitu *Delay* dan *Throughput* serta *Availability* dan *Reliability* pada skenario yang telah ditentukan.

Skenario yang digunakan penulis adalah mengukur performansi jaringan pada Cloud MQTT dan juga pada saat user *me-request* kepada server untuk mengetahui kondisi infus pada saat itu. Dengan pengambilan data sebanyak 30 setiap 15 menit, dan juga pengambilan data secara acak dari 10 akun untuk mengetahui *Availability* dan *Reliability*.

3.4.1 Analisis Reliability dan Availability pada Alat

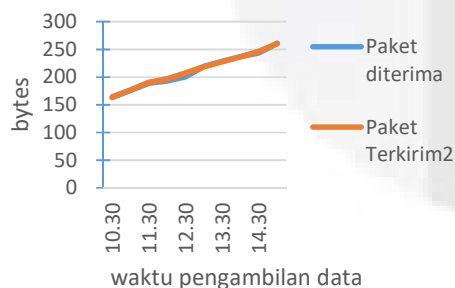
Pada pengujian untuk *Reliability*, penulis melakukan pengujian setiap 15 menit dalam waktu kurang lebih 6 jam, untuk pengambilan data sebanyak 30 kali dan secara berurut setiap 15 menit dengan 3 user yang berbeda dan waktu pemanggilan secara bersamaan. Jika melihat dari hasil data yang

di dapat oleh penulis, sistem ini memiliki kekurangan dalam pendeteksian sensor maupun pengiriman data. Dengan hasil pengujian *Reliability* pada sistem atau keandalan pada sistem ini senilai 87% pada keandalan sistem mendeteksi pergerakan tetesan infus dan 83% pada pengiriman status infus pada *user*.

Pada pengujian untuk menganalisis *availability* pada alat penulis mengambil data dengan cara mengambil data secara acak selama berturut-turut dengan 3 akun yang berbeda, baik secara bersamaan maupun berbeda. Jika melihat dari hasil data yang di dapat oleh penulis, sistem ini memiliki kesediaan dalam pendeteksian sensor maupun pengiriman data. Dengan hasil pengujian *Availability* pada sistem atau keandalan pada sistem ini senilai 100% pada keandalan sistem mendeteksi pergerakan tetesan infus maupun pada pengiriman status infus pada *user*.

3.4.2 Analisis Performansi Jaringan Pada Alat

Pusat komunikasi ada di MQTT *broker*, *broker* ini bertanggung jawab atas terkirimnya semua pesan termasuk jalur distribusinya. Setiap *client* yang mengirim pesan ke *broker*, termasuk jalur distribusinya. Setiap *client* yang mengirim pesan ke *broker*, termasuk juga mengirimkan *topic* ke dalam pesan tersebut. *Topic* merupakan bagian dari *routing information* untuk *broker*-nya. Tiap klien yang menginginkan menerima pesan, bisa *subscribe* ke suatu *topic* tertentu dan *broker* akan mengirimkan semua message yang cocok dengan pola (*pattern*) *topic* tersebut kepada *client* yang sesuai. Mekanisme tersebut membuat klien tidak perlu untuk tau satu sama lain untuk bisa berkomunikasi, melainkan mereka cukup berkomunikasi menggunakan *topic*.

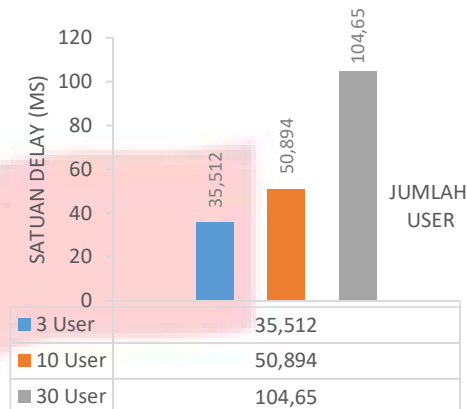


Gambar 4.1 Grafik perbandingan paket yang diterima dan dikirim oleh broker.

Melihat gambar 4.1 dapat dilihat terjadinya packet loss kurang lebih sekitar 0,01% pada saat pengiriman sensor menuju CloudMQTT. Hal ini terjadi dikarenakan adanya masalah koneksi internet pada penulis yang tidak stabil.

Pada penghitungan performansi jaringan yang meliputi delay dan throughput penulis

menggunakan aplikasi wireshark. Wireshark merupakan aplikasi yang dapat mengetahui performansi sesuatu jaringan, dimana membutuhkan IP pada User serta aplikasi. Dengan menggunakan CMD dengan perintah *netstat* penulis dapat mengetahui IP User serta aplikasi line yaitu 10.140.43.136 merupakan IP yang terdapat di PC User dan 172.217.194.188 merupakan IP untuk mengakses Line.



Gambar 4.2 Grafik Delay pengiriman antara Server-Line

Perhitungan performansi melakukan 2 kali percobaan dengan masing-masing data yang diambil setiap 15 menit dalam waktu 3 jam. Sesuai dengan standart ITU-T seperti pada tabel 3.3, hasil yang didapat pada percobaan memiliki rata-rata waktu pengiriman kurang lebih 0,0048 *Second* dan memiliki delay yang masih dapat di kategorikan sebagai *Excellent* karna dibawah 150 ms. Penulis berasumsi bahwa pengiriman dapat dilakukan secara cepat karna adanya koneksi internet yang mendukung dan juga dikarenakan Heroku menggunakan jaringan lokal yang membuat data cepat terkirim.

Jika melihat hasil grafik throughput, tidak stabilnya bentuk grafik dikarenakan terjadinya kendala pada internet yang kecepatan tidak stabil dan paket yang dikirimkan tidak semuanya sama. Pada saat throughput mencapai puncak tertingginya itu dikarenakan pengiriman paket saat waktu internet sedang stabil dan jalur sedang sepi. Paket diterima dalam satuan Kbps. Hasil throughput yang tidak stabil juga dikarenakan adanya gangguan pada penulis yang mengakses internet tidak hanya untuk melakukan uji performansi.

4. Kesimpulan

Setelah dilakukan pengujian dan analisa terhadap sistem yang dilakukan dapat disimpulkan bahwa :

1. Sistem pengiriman status infus pada sensor IR dihubungkan dengan arduino menggunakan protokol MQTT dan dihubungkan dengan media komunikasi Line Bot.

2. Hasil pengujian pada Line Bot dengan mengirimkan pesan "Status Infus" yang langsung terhubung pada Heroku Dyno Bot mengirimkan data secara *Real Time* yang dapat dilihat pada *websocket* CloudMQTT.
3. Perhitungan paket yang dikirimkan antar Sensor dengan CloudMQTT memiliki rata-rata sebesar 212,709 bytes. Pada pengiriman ini memiliki 0,01% *Packet Loss* dalam waktu pengambilan data setiap 30 menit. Hasil yang di dapat penulis dapat dikatakan memiliki nilai yang kecil dan masih dapat digunakan dikarenakan paket tetap dapat diterima.
4. Dengan hasil pengujian *Reliability* pada sistem atau keandalan pada sistem ini senilai 87% pada keandalan sistem mendeteksi pergerakan tetesan infus dan 83% pada pengiriman status infus pada *user*, sedangkan *Availibility* memiliki hasil 100%.
5. Hasil ini di dapat karna adanya kekurangan pada sensor yang hanya dapat mendeteksi jika warna cairan infus sangat pekat, dan terkadang jika infus terkena atau menyentuh tangan mendeteksi adanya halangan.
6. Pada uji pertama untuk menghitung performansi jaringan, penulis mendapatkan hasil rata-rata delay sebesar 22,9 ms. Sementara untuk hasil uji kedua mendapatkan nilai 33,109 ms. Secara hasil yang di dapat dan disesuaikan pada standarisasi maka hasilnya sangat bagus karna kurang dari 150 ms.
7. Heroku Dyno Bot yang memiliki local host menjadikan sistem pengiriman kepada Line Developer memiliki delay yang kecil.
8. Pada uji pertama untuk menghitung performansi jaringan, penulis mendapatkan hasil rata-rata *Throughput* sebesar 26,49 ms. Sementara untuk hasil uji kedua mendapatkan nilai 22,834 ms.

Daftar Pustaka

- [1] Amazon Web Service , [Online]. Available: <https://aws.amazon.com/id/iot/>. [Diakses 14 October 2018].
- [2] "Wikipedia," [Online]. Available: https://id.wikipedia.org/wiki/Arduino#cite_note-1. [Diakses 14 October 2018].
- [3] E. Pr., "Mengenal MQTT Protokol IoT," Medium, 6 Oktober 2015. [Online]. Available: <https://medium.com/pemrograman/mengenal-mqtt-998b6271f585>. [Diakses 2018 Oktober 19].
- [4] A. Rohma. [Online]. Available: <https://halosehat.com/review/tindakan-medis/jenis-jenis-cairan-infus>. [Diakses 23 September 2018].
- [5] D. Kurnia, "okezone.com," 12 November 2015. [Online]. Available: <https://lifestyle.okezone.com/read/2015/11/12/481/1248091/penyembuhan-jantung-koroner-dengan-infus>.
- [6] "alodokter," [Online]. Available: <https://www.alodokter.com/dasar-dasar-prosedur-memanfaatkan-cairan-infus>. [Diakses 1 Oktober 2018].
- [7] N. S. Sachchidanand Singh, "Internet of Things(IoT): Security Challenges, Business Oppurtunities & Reference Architecture for E-Commerce," dalam *International Conference on Green Computing and Internet of Things (ICGCIOT)*, Noida, India, 2015.
- [8] Admin, Business of Apps, 2016 July 25. [Online]. Available: <http://www.businessofapps.com/15-internet-things-facts-know/>. [Diakses 14 October 2018].
- [9] "Arduino," [Online]. Available: <https://www.arduino.cc/en/Guide/ArduinoMega2560>. [Diakses 14 Oktober 2018].
- [10] "NyebarIlmu," 5 January 2018. [Online]. Available: <https://www.nyebarilmu.com/cara-mengakses-module-sensor-warna-tcs230-menggunakan-arduino/>. [Diakses 14 Oktober 2018].
- [11] Sinauarduino, "Modul Wifi ESP8266," Redaksi Sinauarduino, 6 April 2016. [Online]. Available: <http://www.sinauarduino.com/artikel/esp8266/>. [Diakses 17 Oktober 2018].
- [12] S. P. Aji, "ALAT MONITORING TETESAN INFUS MENGGUNAKAN WEB SECARA ONLINE BERBASIS ESP8266 DENGAN PEMROGRAMAN ARDUINO IDE," 2017.
- [13] R. P. H. N. Hudan Abdur Rochmar, "Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada Smarthome," *Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. I, pp. 445-455, 2017.
- [14] N. Shofa, Pemantauan Penggunaan Infus Berbasis WiFi (802.11) melalui Protokol MQTT, Bandung, 2017.
- [15] N. Purnamasari, "Rumus-Rumus Hitungan Dalam Keperawatan," 29 Desember 2015. [Online]. Available: <https://ninniepurnamasari.wordpress.com/2015/12/29/rumus-rumus-hitungan-dalam-keperawatan/>. [Diakses 24 Oktober 2018].